This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS

- TEXT CUT OFF AT TOP, BOTTOM OR SIDES

- FADED TEXT

- ILLEGIBLE TEXT

- SKEWED/SLANTED IMAGES

- COLORED PHOTOS

- BLACK OR VERY BLACK AND WHITE DARK PHOTOS

- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(51) International Patent Classification 6 :		(11) International Publication Number: WO 95/05712
H04L	A2	(43) International Publication Date: 23 February 1995 (23.02.95)

(54) Title: SECRET KEY EXCHANGE

(57) Abstract

    The present invention describes a method for enabling users of a cryptosystem to agree on secret keys. In one embodiment, a trusted agent chooses at least one individual key for each user, with at least a portion of such individual key being secret. At least some of the individual keys are then stored in physically secure devices, and the pair of users i and j use their individual keys to compute a common secret key. In another embodiment, each trustee of a group of trustees chooses at least one individual key for each user, with at least some portion of such individual key being secret. The keys chosen by a sufficiently small number of such trustees, however, are insufficient for computing the common secret key of the users. Other hardware and software key exchange protocols based on these two techniques are also disclosed.

SECRET KEY EXCHANGE

The present invention relates generally to
secure message techniques and more particularly to a
method for enabling users of physically-secure
devices to agree on a private key.

5   Conventional cryptographic algorithms allow two
users, who already possess a common secret key, to
exchange private messages even when communicating
over a public network. Such systems possess very
fast software implementations, inexpensive and fast

10   hardware implementations, and, most importantly, are
very secure. In fact, their security simply relies
on one-way functions: functions f that are easy to
evaluate but hard to invert, that is, for which it
is hard, given a generic value $z=f(x)$, to find any

15   value y such that $f(y)=z$.

Despite these main advantages, conventional
cryptosystems, however, are not very useful. Prior
exchange of a common secret key (e.g., by physically
meeting in a secure location) with every person with

20   whom one wants to talk to in private is, to say the
least, cumbersome in most scenarios.

To overcome this difficulty, several methods
have been developed to allow two people to agree on
a common secret keys in a convenient manner.

25   Unfortunately, however, until now all publicly known
protocols for this task are either based on the
assumed computational difficulty of a given number
theoretical problem (as in the Diffie-Hellman
algorithm and the RSA algorithm), or they rely on a

30   non-realistic amount of trust.

In the case of RSA, the encryption function
$f(x)$ typically is x^e mod n, where n is a
publicly-known product of two large prime integers
p_1 and p_2 (known only to the user who publishes n

35   and e), and e is a publicly known exponent
(relatively prime with p_1 and p_2). In the RSA

system, if a user X publishes two values e and n as above, then user Y can select a secret key k in an arbitrary manner and communicate it privately to X, by looking up X's publicized values, computing $k' =$
5   k^e mod n, and sending k' to X over a public network. If computing e-roots modulo a composite integer whose factorization is not known is a virtually impossible computational problem, then only user X will be capable of retrieving k from k';
10   in fact, only X knows $n's$ factorization (i.e., p_1 and p_2), and this knowledge makes extracting e roots feasible, though not trivial.

In the case of the Diffie-Hellman scheme, two users X and Y, respectively possessing public keys
15   g^x mod p and g^y mod p (where p is a prime integer and g a generator mod p), and corresponding secret keys x and y, agree on a common secret key g^{xy} mod p as follows. User X computes a value $g^{yx} = (g^y)^x$ mod p (which he can do because he knows Y's public key
20   and his own secret key); user Y computes $g^{xy} - (g^x)^y$ mod p (which she can do because she knows X's public key and her own secret key. Since multiplication is commutative, $g^{yx} = g^{xy}$ mod p is the desired common secret key.

25   In both the RSA and the Diffie-Hellman algorithms, however, the operations involved for secret-key exchange are quite time-consuming in software (computations of the type a^b mod c are not-trivial whenever these values are large), or
30   they require complex and expensive VLSI chips for fast modular exponentiation. Thus, building large-scale systems for secret-key exchange using such techniques would require a great financial investment.

35   More importantly, the assumptions necessary for the above secret-key exchange schemes to be secure

-3-

are very rigid. In the case of RSA, secret-key
exchange is performed by means of an encryption
function, $f(x) = x^e$ mod n, that should not simply be
one-way, but should also possess a secret (i.e., the
5   factorization of n) knowing which inverting f (i.e.,
computing x from $f(x)$) should become possible rather
than practically impossible. While it is widely
believed that one-way functions exist, fewer
researchers believe that one-way functions possess
10   this additional property. Similarly, in the case of
Diffie-Hellman, g^x mod p not only needs to be
one-way, but it should also possess additional
algebraic and multiplicativity properties. Again,
few people believe that one-way functions satisfying
15   such additional algebraic constraints exist.
Indeed, continuous algorithmic advances are made
that make factoring integers and solving the
discrete logarithm problem easier.
         Therefore, conventional cryptography does not
20   provide any efficient means to achieve secret-key
exchange.
         Other algebraic schemes for secret-key exchange
have been devised by Blom and by Blundo et al., but
these schemes rely upon an unrealistic amount of
25   trust. In fact, not only do these schemes require a
central authority that knows all the individual
secret keys of the users, but also that essentially
that all of the users in a large system are
trustworthy. For instance, in Blom's case, as
30   described in an article titled "An Optimal Class of
.Symmetric Key Generation Systems," *Advances in
Cryptology: Proceedings of Eurocrypt 84, Lecture
Notes in Computer Science*, Vol. 209,
Springer-Verlag, Berlin, 1987, pp. 335-338, a
35   trusted authority prepares and distributes keys to a
group of n users. If each user key is $B \cdot k$-bit

SUBSTITUTE SHEET (RULE 26)

long, the authority can compute from his own key
(without interaction) a k-bit long common secret key
for every other user in the system. All these keys
will remain secret, unless k of the users
5 collaborate and reveal to each other the keys in
their possession. If this happens, they can compute
the secret keys of every other user in the system.
     Moreover, with such schemes few bad users may
achieve the same results of many more bad users by
10 forcing good ones to surrender their own secret
keys. While in other schemes forcing some users to
reveal their own keys may allow an enemy to
understand at most the communications of those users
(who will be aware of having lost privacy), in these
15 algebraic schemes an enemy who has forced a
sufficient number of users to reveal their own
secret keys will understand the communications of
all users, which is obviously untenable.
     In sum, therefore, prior art techniques are
20 inadequate for setting up truly viable secret-key
exchange systems, especially where such systems are
designed for large-scale use where the number of
potentially dishonest users is enormous and there is
no single individual that all users would trust to
25 know their keys.
     It is therefore a principal object of the
present invention to provide for new secret-key
exchange protocols for users of crytosystems.
     It is another principal object of this
30 invention to devise secret-key exchange schemes that
remain practical even on a national scale, and when
there is no single individual or entity that all
users would trust to know their keys.
     It is a further object of the invention to
35 provide novel secret-key exchange schemes that rely
on conventional cryptographic tools, such as one-way

functions without any further property, or secure
hardware. Secure hardware has a portion that is
tamperproof, and thus the information contained in
it may be destroyed but is otherwise unreadable and
5 untamperable.

It is yet a further object of the present
invention to provide new and alternative algorithms
for key exchange that are not based on the assumed
difficulty of some particular number theoretic
10 problem and which are thus immune from any
algorithmic advances that may be made in the future.

It is still another important object to provide
key exchange techniques that are particularly
well-suited for use with secure hardware and that
15 are very fast and very economical to implement.

It is another object of the invention to make
use of trustees for turning conventional
cryptosystems into secret key exchange protocols.
According to the invention, a mechanism is provided
20 by which two users i and j can compute a secret pair
key $K_{i,j}$ that is known only to themselves (and, if
desired, trusted agents). This key is then used as
a private key to encrypt a randomly-generated
session key which is used to encrypt the session
25 traffic using a secure hardware chip. Each solution
has the property that the pair key can be computed
by each party without prior communication between
the parties. For example, user i can compute $K_{i,j}$
without communicating with user j. Similarly, the
30 receiver of a message can compute $K_{i,j}$ without
additional communication with the sender. The first
two solutions are hardware-based in the sense that
they rely on the difficulty of extracting secret
keys from the protected memory of a chip. These
35 schemes are parameterized in terms of the number of
chips that need to be "opened" before an adversary

-6-

can successfully eavesdrop. The third solution is
software-based and it does not rely on protected
memories or tamper-proof chips. This scheme relies
on trustees, only some of whom need to be honest.

5   The foregoing has outlined some of the more
pertinent objects of the present invention. These
objects should be construed to be merely
illustrative of some of the more prominent features
and applications of the invention. Many other

10 beneficial results can be attained by applying the
disclosed invention in a different manner or
modifying the invention as will be described.
Accordingly, other objects and a fuller
understanding of the invention may be had by

15 referring to the following Detailed Description of
the preferred embodiment.

         In April 1993, the Clinton Administration
announced its intention to develop and produce
cryptographic hardware for widespread use within the

20 government. The hardware, or so-called Clipper
Chip, is designed to provide secure communication
after common secret keys have been established, but
the announcement neither dealt with nor suggestion a
solution to the problem of secret-key exchange. The

25 Clipper Chip will be made using a special VLSI
process which is designed to prevent reverse
engineering. In particular, the encryption and
decryption algorithms used in the Clipper Chip will
be classified, but the chip itself will not be

30 classified. Each Clipper Chip will also contain a
protected memory for secret keys, and such products
are currently known in the industry. The protected
memory is designed to prevent anyone (even the
legitimate user of the chip) from gaining full

35 access to the keys contained therein.

According to the announced specification for
the Clipper Chip, each chip i will be equiped with a
unique secret K_i that is formed by a irreversible
process from two pieces of the secret key $K_i^{(1)}$ and

5   $K_i^{(2)}$. The pieces of the secret keys will be held
by system-wide trusted agents T_1 and T_2. Actually,
only one of the agents needs to be trusted since T_1
will hold only the first piece of each secret key
and T_2 will hold only the second piece. When two

10 parties wish to communicate using the new system,
they first agree, in some way, on a session key S
and then they enter this key into their respective
chips. This key is used by the chips as an
encryption/decryption key for the message traffic.

15 In other words, once the session key is selected,
the Clipper Chips function as a conventional
private-key cryptosystem, but with the following
difference. The Clipper Chips also transmit the
session key S being used in encrypted form using the

20 secret key for the chip, thereby allowing trusted
agents to eavesdrop on the conversation. The reason
for transmitting the session key in this fashion is
to preserve court-authorized eavesdropping. In
fact, when presented with a legitimate court order

25 for eavesdropping the communications of a suspected
user who encrypts his messages by means of Clipper
Chip i, each of the two trustees will reveal the
share of K_i in its possession, so that the Police
will be able to reconstruct K_i, the session key and

30 thus the message of the suspected user.
          In a first embodiment of this invention,
referred to hereinafter as a hardware-based
"trusting" scheme, it is assumed that there is a
trusted agent T who wants to enable a group of N

35 users (named 1 through N) to communicate privately
among each other by using k-bit secret keys,

assuming that no more than B of these N users will
cooperate for the purpose of discovering the others'
secret keys. On input N, B, and k, the trusted
agent T chooses p to be a prime greater than N and k
5   and randomly selects a B-degree bivariate polynomial
$F(x,y)$ mod p. That is, she selects $(B + 1)^2$ random
elements modp. $F(x,y)$ is thus known to the trusted
agent and consists of $k(B + 1)^2$ bits. The trusted
agent T derives from main secret $F(x,y)$ the
10 individual key of every user. In fact, the
individual secret key assigned by T to user i
consists of the two univariate polynomials P_i =
$P_i(y) = F(i,y)$ and $Q_i = Q_i(x) = F(x,i)$. P_i and Q_i
constitute the secret key of chip i, while the
15 number i is the public key of user i.

These individual secret keys allow two users i
and j, $i < j$, to easily agree on a common secret key
K_{ij}; namely, $K_{ij} = F(i,j)$. This value is computed
by user i evaluating the secret polynomial P_i at
20 point j, and it is computed by user j evaluating the
secret polynomial Q_j at point i.

An adversary who has opened at most B chips
cannot predict, in an information-theoretic sense,
the secret pair of any two users. In this "trusting
25 scheme" embodiment, the secret key of each user is
$2k(B+1)$ bits long, while the corresponding public
key consist of log N $(<k)$ bits. Thus, the number of
bits that must be remembered by each user depends on
B but not on the total number of users N.
30 Therefore, for the scheme to be practical
(notwithstanding the difficulty of the trusted
agent), the upper bound on the number of bad users
should be reasonably small. For instance, if it is
assumed that there are at most B = 100 bad users in
35 the system, and that k = 100 is the common-key
length, then it suffices that each user remember

20,200 bits, which should be acceptable in some
scenarios.

The "trusting" scheme is improved according to
the invention to remove the concern that, at most, a
5   few users may be bad. This alternate technique also
uses secure hardware. At the outset, it is assumed
that every user in the system communicates privately
by means of a secure chip, like the Clipper Chip.
The trusted authority T computes for each user i an
10   individual key K_i like in the trusting scheme or any
other scheme that allows any pair of users i and j
to agree on a common sufficiently secret key K_{ij}.
Preferably, such a scheme should be one wherein
knowledge of less than a given number (B>0) of
15   individual keys does not enable one to understand
the communications of other users. The trusted
authority does not, however, give individual key K_i
to user i, rather the authority stores K_i in a
protected portion of a secure chip (chip i). The
20   authority then gives chip i to user i. In this
embodiment, no user knows his own key. Nonetheless,
every user i can privately communicate to every
other user j. In fact, chips i and j knowing,
respectively, the individual secret keys of users i
25   and j, can internally compute their common secret
key K_{ij}, and thus encrypt and decrypt messages with
that key, as will be seen. While the decrypted
messages will be output so as to make them readable
to the recipient, it is preferable that the
30   operations needed for computing the common key K_{ij}
will take place in the protected portion of the
chips. Such a technique has advantages over the
trusting scheme because individual keys are
unavailable to the B users.
35   At this point, however, the system still needs
to trust a central agent. This last difficulty can

also be avoided according to a so-called
"non-trusting" scheme. For simplicity, assume there
are two trustees, T_1 and T_2, each acting as the
central authority of the trusting scheme. Thus T_1
5   will choose individual keys $K^1_1,\ldots,K^1_N$ for the N
users, and trustee T_2 will independently choose
individual keys $K^2_1,\ldots,K^2_N$. The key assigned to
chip i is then be a combination of K^1_i and K^2_i; and
such combination may be their concatenation or their
10 bit-by-bit sum modulo 2.

According to the invention, precautions must be
taken so that only chip i knows K_i. Thus, it is
preferred to have each trustee t store K^t_i into chip
i directly. For instance, chip i is first
15 physically brought to trustee T_1 (or to some
facility controlled by it) so that T_1 can privately
store K^1_i into the chip, and then chip i is sent to
trustee T_2, who will privately store its own value
K^2_i into the chip. In this way, no one will learn
20 both K^1_i and K^2_i. In fact, once one of these values
is stored into chip i, it will be unreadable by
anyone. It will then be chip i itself to combine
K^1_i and K^2_i internally so as to compute its own
individual key K_i.

25   Alternatively, each trustee transmits its own
share of key K_i to the chip encrypted so that only
chip i can understand it. For instance, chip i may
generate its own public encryption key, known in
particular to the trustees, and keep its own
30 decryption key inside and thus protected. In this
manner, there is no need that each trustee be
physically provided every chip in order to store in
it his own share of the individual key. In yet
another alternative, each secure chip is
35 manufactured so that it will generate its own
individual key inside, compute two shares of this

key, and inform each trustee of its own share. Preferably, this will be done in a secure manner, e.g., by having the chip transfer the shares encrypted, so that only the right trustee can
5 understand its own share.

It should be noticed that in this hardware-based "non-trusting" scheme not all trustees must be trusted. In fact, for the scheme to be secure it is enough that at least one of them
10 be trusted. Moreover, since trustees will be selected so as to be very trustworthy, the chance that at least one of them will be trustworthy should be greater than the chance, in the trusting scheme, that at most B users are bad.

15   Although the Clipper Chip announcement envisages a scenario with trustees and secure chips, such techniques use them merely to provide fair but conventional private-key cryptosystems, i.e., it enables private communication (wiretappable in some
20 proper cases) only between users who have already established a common secret key. To the contrary, the present invention uses the trustees and secure hardware in conjunction with other protocols to allow a pair of users to establish a common secret
25 key.

In addition, it should be noticed that the non-trusting scheme, beyond providing a secret-key exchange protocol based on conventional methods, also guarantees court-authorized eavesdropping. In
30 fact, after being presented with a legitimate court order, the trustees may release their own piece of the individual key of a suspected user i, so as to allow the police to monitor his communications. For further security, the trustees might disclose their
35 own shares to a secure chip, used by the police to monitor the conversations of suspected users, so

that no one, except the monitoring chip itself, will
know the suspected user's individual secret key.

Both the trusting and non-trusting schemes can
be adapted to work in a scenario where there are
5   various gradations or "hierarchies" of security.
For instance, assume that the users are categorized
in S security levels, 1,2,...,S, where level 1 is
the highest level of security, and level S is the
lowest level of security. Then, assume it is
10   desired to provide a scheme for secret-key exchange
that satisfies the two properties. First, users of
different security levels can directly talk to one
another (preferably using the same hardware and/or
scheme). Second, a conversation between two users
15   always takes place at the highest common level of
security. In particular, in order for an adversary
to eavesdrop on a conversation between a user of
security level p and a user of security level q,
where p<q, the adversary will need to open
20   sufficiently many chips of level q or better, and
opening chips at a lower level of security is of no
use at all. The reference to security "hierarchies"
herein does not mean that some users have better
encryption algorithms or that people of higher
25   security levels can spy on those with lower ones.
Rather, the term means that, although every user can
talk to any other user by the same scheme,
compromising the keys of users of lower security
does not help in computing the keys of users at
30   higher levels.

Encryption schemes with different security
levels are most important, particularly in a
secure-chip scenario. In fact, there are likely to
be fewer chips at the higher security level, and
35   since they are likely to be guarded more closely, it
will be much more difficult for an adversary to

obtain such chips, and he will have to open more of
them before being able to eavesdrop (since we can
replace N by the number of users with that level of
security, which is smaller).

5   The above properties are provided according to
the present invention. In one embodiment, new
individual keys using S independent set of
individual keys as in any of the above-described
schemes (i.e., both in a trusting and non-trusting

10 scenario), are arranged so that the chips of
security level S work as before, while the chips of
higher security levels have more individual secret
keys. For instance, let i be a chip of security
level S. Then, i will securely carry an individual

15 secret key, K^S_i. If the particular bivariate
polynomial scheme is used to compute the individual
keys as in the trusting case, $K^S_i = F_S(i,y), F_S(x,i)$;
where $F_S(x,y)$ is a bivariate polynomial assigned to
security level S. If, instead j is a chip of

20 security level S - 1, then it securely carries a
secret individual key consisting of two keys: K^S_j
(i.e., its individual key as it were a chip of
security level S) and a second "S - 1 level" key
K^{S-1}_j. In particular, if the trusting scheme is

25 used (with the specific bivariate polynomial method)
$K^{S-1}_j = F_{S-1}(j,y), F_{S-1}(x,j)$; where $F_S(x,y)$ is a
bivariate polynomial assigned to security level S -
1. Thus a chip z of level q will have an individual
key consisting of S + 1-q individual keys, one for

30 each level l between q and S. When a chip of level
p wishes to communicate to a chip of level q, and
p<q, then they may do so by using their q-level
individual keys. Or, they may communicate by using
all their individual keys from level S to level q.

35 e.g., by means of a common secret key obtained by

combining their common secret keys relative to
security levels S through q.

Different security levels allow for a great
deal of convenience. In an army, for instance,
5   security level may grow with rank. In this way, any
officer can still have a direct and private
conversation with every soldier, but even if many
ordinary soldiers are captured and their chips are
successfully opened by the enemy, the enemy will
10 never be able to eavesdrop conversations between
officers.

Also, the present scheme makes it harmless to
export cryptographic hardware and share
cryptographic systems with foreign countries.
15 Assume in fact that our system were adopted by a
country C, so that national chips were given a
higher security level than that of exported chips.
Then, even if all foreign countries actively tried
to open the encryption chips given to them, and
20 succeeded in doing so for sufficiently many chips,
no harm would result for C's national security. In
fact, no one could eavesdrop conversations between
two chips of country C.

According to still other embodiments of the
25 invention, it is desirable to use secure chips along
with conventional one-way functions. As before,
assume N is the total number of users, B is the
maximum number of chips openable by an adversary and
k is the number of bits in each secret pair key
30 (e.g., k=100). This aspect of the invention is
implemented with a trusted agent (which computes
special information for the users) or by relying
instead on a group of "only moderately trusted"
trustees.
35   In the trusted agent method, there are M secret
keys $X_1,\ldots,X_M$ each with k bits that are generated

at random and kept by the trusted agent. This agent
also generates a public and private key for each
user of the system as follows. The public key for
user i will consist of M integers $a_1,\ldots,a_M$ selected

5   at random from the interval [1,L], where L is a
parameter of the system. The secret key stored in
the unopenable portion of user i's chip consists of
M (preferably k-bit long) strings $Y_1,\ldots,Y_M$ where Y_m
= $h^{a_m}(X_m)$ for $1 \leq m \leq M$, and h is a suitably-chosen

10   one-way function. For simplicity, h is considered
an ideal one-way function that maps any input string
x to a random, preferably k-bit long, output h(x).
Also, one one-way function h is used for simplicity
only as more than a single one-way functions may be

15   used (i.g., one could use a different one-way
function h_j for each value X_j). When user i wants
to send a message to user j, he does so by selecting
a session key at random, which key is encrypted
using a pair key $K_{i,j}$ that is generated as follows.

20   Let $a_1,\ldots,a_M$ be the public key for user i and
let $\mathfrak{t}_1,\ldots,\mathfrak{t}_M$ be the public key for user j. Then
chip i computes $s_1,\ldots,s_M$ where s_m = $\max(a_m,\mathfrak{t}_m)$ for
$1 \leq m \leq M$, and the chip combines the values
$h^{s_1}(X_1),\ldots,h^{s_m}(X_M)$ into a single value $K_{i,j}$. For

25   instance, the chip sets
$$K_{i,j} = h(h^{s_1}(X_1)\acute{A}\ldots\acute{A}h^{s_m}(X_M)),$$
Where the symbol $\acute{A}$ denotes concatenation and h is,
for simplicity only, the same function used above.
        Note, therefore, that both chip i and chip j

30   can compute $K_{i,j}$ given their own secret key and the
public key of the other, thus allowing users i and j
to have a private communication. Though the common
secret key $K_{i,j}$ (if properly generated as a
combination of the values $h^{s_1}(X_1),\ldots,h^{s_m}(X_M)$) could

35   be known to users i and j, it is preferred that it
be known only to their chips.

SUBSTITUTE SHEET (RULE 26)

In this embodiment, the length of the public
key for each user is about M log L bits, although
this length can be reduced to log N bits if the
public key for user i is generated using a public

5   pseudo-random number generator with a seed based on
i. For example, we might use h(i) as the public key
for user i. The length of the secret key for each
user is Mk bits, although the trusted agent may
possess a much more compact representation, if it

10 generates the users' secret keys in an algorithmic
manner.

The above scheme can be modified for use with
two or more trusted agents. For instance, it is
desirable to use a set of t independent trustees,

15 each one of which acts as the trusted agent such as
described above. In this case, the overall common
secret key key $K_{i,j}$ is a combination of t keys: the
$K_{i,j}$-key relative to each trustee. This
modification does not affect the security and the

20 relevant properties of the trusted agent scenario.

The previous schemes also can be easily
modified for use with security hierarchies. Let the
users be categorized into S security levels
$1,2,\ldots,S$, where again level 1 is the highest level

25 of security, and level S is the lowest level of
security. Then, the invention implements the same
scheme as described above except that the public key
for a user at security level q is selected so that
a_m is a random integer in the range $[1 + (q -$

30 $1)L, qL]$ for $1 \leq m \leq M$.

This modification does not increase the storage
requirement of chips with higher security level, but
increases the time to compute a pair key by a factor
of at most S, since chips belonging to users of

35 higher security level will have to iterate the
underlying one-way function more times. The

SUBSTITUTE SHEET (RULE 26)

modified scheme, however, allows all users (no
matter what their security level) to utilize the
same type of hardware, thus decreasing the price of
the chips. It also allows users of different
5   security levels to talk directly to one another in a
secure and transparent way (i.e., the chips will
perform the same type of operations, independently
of the security levels involved). It also allows
the conversation between the two users to always
10 take place at the highest common level of security.
     While secure chips are manufactured so that
they cannot be "opened" at all, it is still desired
to guard against the possibility that an adversary,
with enough effort and money, can read a few bits
15 from a chip before destroying it. If it is assumed
an adversary were only capable of reading 5 bits
from a secure chip before destroying it, then, in
the non-trusting scheme, he would have to open
Bk/5>B chips before eavesdropping any conversation
20 (i.e., 2,000 chips if k - B = 100). In the one-way
function scheme, however, the invention forces the
adversary to open many more chips by simply
increasing L. This is because learning 5 bits of
$h^n(X_m)$ for many different n will be of little help
25 to the adversary. By making L large, chips with the
identical portions of the secret key will be few and
far between. Increasing L also increases the time
for a legitimate user to compute a pair key, but
computing with one-way functions is very fast and
30 can be made fairly large without difficulty.
     According to a further feature of the
invention, there is now described a "software-based"
scheme for exchanging keys. This solution does not
rely on any protected hardware at all. Starting
35 again with the trusted agent scenario, there is a
single secret key K which is held by the agent. The

-18-

secret key for the ith user is $K_i = h(K\acute{A}i)$. There
is no public key for the ith user. There is a
public pair key for each pair of users i and j,
however, which is a simple and secure combination of

5   their individual secret keys; for instance,

$$t_{i,j} = h(K_j\acute{A}i)\hat{e}h(K_i\acute{A}j).$$

By "simple and secure combination" it is deemed to
mean that with knowledge of the public pair key and
any of its two individual keys, the other individual

10   key may be easily computed, but given the public
pair key only the two individual keys cannot be
easily guessed.

     The secret pair key used by i to send a message
to j is $K_{i,j} = h(K_j\acute{A}i)$. User i computes this key by

15   retrieving the public pair key $t_{i,j}$ from his
personal directory (or from the public directory)
and then computing $t_{i,j}\hat{e}h(K_i\acute{A}j)$. In fact, user i
knows the identity j of the user he wants to talk
to, and his own secret key K_i. This key $K_{i,j}$ is

20   then used to encrypt a session key or a message
directly. To read the message user j simply
computes $h(K_j/_i)$. No table lookup is needed on the
recovering end since j knows his own key K_j and may
easily learn the identity of the sender -- e.g., i

25   may send his own identity in the clear together with
his encrypted message.

     It is preferable that the trusted agent
computes the individual user keys algorithmically,
preferably, as a function of a "master key" K, as

30   indicated above. However, the particular choice of
$K_i = h(K\acute{A}i)$ is just one of the many ways to generate
K_i algorithmically. It is also possible, however,
to have the trusted agent choose K_i in other secure
ways; in particular, purely at random.

35   It should also be appreciated that the
individual secret keys of the present scheme can

also be stored by the trusted agent directly in secure chips, rather than given to the users. Alternatively, the users' individual keys can be stored in secure chips by the users themselves.

5 There is, however, less reason to do so. In fact, a main advantage of the present scheme over the previous ones is that if an adversary learns the individual keys of a set of users (whether by corrupting them or by opening somehow their secure

10 chips), he can only decipher communications involving that set of users. In other words, knowledge of K_i or $h(K_i Áj)$ can only be used to recover information about $h(K_j Ái)$, which is only useful for communication with user i.

15   The scheme just described is, moreover, very economical to implement. The only potential disadvantage is that there are a potentially large (e.g., N^2) number of public pair keys. This disadvantage can be overcome by publishing a

20 personalized directory of public pair keys for each user. In other words, each user can keep his own phone book of k-bit keys.. Alternatively, each user i can simply call up the trusted agent (using a 411-like phone call, for example) each time he wants

25 to communicate with a new user j. The trusted agent, who knows the secret key K can then quickly recompute the public pair key, $t_{i,j}$; for the two users and send it to i. In such case the trusted agent does not need to store all the N^2 pair keys to

30 answer these calls, because he can recompute any pair key instantly from his secret value K. This is a main advantage of choosing the individual user keys algorithmically as a function of KÁ Also, the trustee is only giving out public information in

35 satisfying the request by sending $t_{i,j}$. Thus, there is no need to use encryption or special precautions

for this transmission. In this embodiment there is
also no specific need for authentication, since any
user z ℞ i,j may request and obtain the public pair
key $t_{i,j}$ without any harm to i's and j's privacy.

5   For these reasons, the call-up mechanism is
particularly attractive. Indeed, the whole process
can be easily automated -- a user can dial in the
identity of the two users i and j and then receive
the 10-byte public pair key in response. Thus the
10   effort needed to call up someone privately for the
first time is at least as efficient as finding out
the number of a user that is called for the first
time in an ordinary telephone network.

Further, once a user obtains a public pair key,
15   he can store it locally (electronically) so that if
and when he calls that party again, the pair key can
be retrieved automatically without a call to the
trusted agent. Of course, a user that is on the
receiving end of a message never needs to call the
20   trusted agent for any information, since the
receiver can trivially reconstruct the pair key.

The software-based scheme also works with
multiple trusted agents, only one of whom needs to
be honest. One approach is to have each trustee
25   act, independently, as the trusted agent so that
there will be an individual secret key K^t_i for each
user i and trustee t, and a common secret key $K^t_{i,j}$
for every users i and j and every trustee t. User i
will send user j messages encrypted with a total
30   common key $K_{i,j}$ which may be obtained by combining
the t common secret keys relative to each trustee.
Because every one of these t keys can be computed by
the sender and by the receiver in any of the ways
explained above, so can their total common secret
35   key. Thus, for instance, if there are two trustees
computing users' individual keys algorithmically and

each trustee t provides user i with a public pair key t^t_{ij} in order to talk privately to user j, then when i wants to talk privately to j for the first time, he must make two phone calls to obtain the
5   necessary information.

     While the whole system can be securely implemented in software, it may be advantageous to make use of secure chips for some other ends. Assume, for instance, that the trustees are
10 government agencies unwilling to provide responses to call-up inquiries. In this case, the trustees can use telephone companies or others to provide the users the necessary public pair keys without making such entities de facto trustees. For instance, each
15 trustee may provide a telephone company with a secure chip which contains in its protected memory the trustee's master secret key or sufficient information to compute the public pair keys. The telephone companies may then use these chips for
20 computing the public pair keys, in response to users' requests in a call-up manner, without ever learning the trustees' master keys, and thus without being ever able to illegally eavesdrop any conversation. The trustees will then be used solely
25 for law-enforcement purposes. For instance, they will, when presented a legitimate court order, provide the secret individual key. Secure chips, however, may not be needed even for this application, for instance, if the trustees provide
30 the phone companies with the complete set of public pair keys.

     It should be appreciated by those skilled in the art that the specific embodiments disclosed above may be readily utilized as a basis for
35 modifying or designing other methods and techniques for carrying out the same purposes of the present

invention. It should also be realized by those skilled in the art that such equivalent methods do not depart from the spirit and scope of the invention as set forth in the appended claims.

5

10

15

20

25

30

35

CLAIMS

1. A method for enabling users of a cryptosystem to agree on secret keys, comprising the steps of:

5   for each user, having a trusted agent choose at least one individual key, at least a portion of which is secret;

having at least some of the individual keys stored in physically secure devices; and

10   having a pair of users i and j use their individual keys to compute a common secret key.

2. The method as described in Claim 1 wherein some of the users belong to different security 15 levels.

3. The method as described in Claim 2 wherein the individual keys of users of lower security levels contain substantially no useful information 20 for computing the common secret key of a pair of users of higher security levels.

4. A method for enabling users of a cryptosystem to agree on secret keys, comprising the 25 steps of:

for each user, having each trustee t of a group of trustees choose at least one individual key, at least some portion of which individual key is secret and wherein the keys chosen by a sufficiently small 30 number of trustees are substantially insufficient for computing the common secret key of the users;

having at least some of the individual keys stored in physically-secure chips; and

having a pair of users i and j use their 35 individual keys to compute a common secret key.

5. The method as described in Claim 4 wherein
some of the users belong to different security
levels.

5   6. The method as described in Claim 5 wherein
the individual keys of any sufficiently small
subgroup of users of a lower security level contain
substantially no useful information for computing
the common secret key of a pair of users of a higher
10 security level.

7. The method as described in Claim 5 wherein
the individual keys of any subgroup of users of a
lower security level are not useful for computing
15 the common secret key of a pair of honest users of a
higher security levels.

8. A method for enabling users of a
cryptosystem to agree on secret keys, comprising the
20 steps of:
          generating at least one public key for each
user by interating at least a conventional one-way
function on at least one secret value; and
          having a common secret key for a pair of users
25 be computable based on information that includes one
user's secret information and the other's public key.

9. The method as described in Claim 8 wherein
some ofthe users belong to different security levels.
30

10. The method as described in Claim 9 wherein
secret information relative to users of a lower
security level is substantially useless for
computing the common secret key of a pair of users
35 of a higher security level.

11. A method, using secure chips, for enabling users of a cryptosystem to agree on secret keys, comprising the steps of:

     having each trustee t of a group of trustees
5 generate at least one public key for each user by evaluating at least one conventional one-way function on at least one secret value;

     having a common secret key for a pair of users be computable based on information that includes
10 secret information of one user and information that includes public information about the other user; and

     having any subset of information generated by a sufficiently small group of trustees be substantially insufficient for computing the common
15 secret key of a pair of honest users.

12. The method as describdd in Claim 11 wherein some of the users belong to different security levels.
20

13. The method as described in Claim 12 wherein any secret information relative to a sufficiently small group of users of a lower security level is substantially useless for
25 computing the common secret key of a pair of users of a higher security level.

14. A method for enabling users of a cryptosystem to agree on secret keys, comprising the
30 steps of:

     generating at least one public key for each pair of users; and

     having a common secret key for a pair of users be computable based on information that includes
35 their own public key and their own secret keys.

15. A method for enabling users of a cryptosystem to agree on secret keys, comprising the steps of:

generating at least one common public key for
5   each pair of users; and

generating at least one secret key per user; and

having a common secret key for a pair of users be computable based on information that includes the common public key for the pair of users and the
10   secret keys of the pair of users.

16. A method for enabling users of a cryptosystem to agree on secret keys, comprising the steps of:
15   having a group of trustees generate a piece of information for each user that includes at least one secret key;

for each user, generating, from the pieces of information relative to that user generated by the
20   trustees, a piece of information that includes at least one secret key and wherein the pieces of information generated by a sufficiently small number of trustees are practically insufficient to compute the common secret key of a pair of users.;
25   generating at least one common public key for a pair of users; and

having a common secret key for a pair of users be computable based on information that includes the common public key for the pair of users and the
30   secret keys of the pair of users.

35